Working Group:
Internet Draft

Document:
draft-bianchi-blefari-end-to-end-qos-02.txt

G. Bianchi
University of
Palermo, Italy

N. Blefari-Melazzi
University of
Perugia, Italy

Category: Informational

November 2001
Expires April 2002

A Migration Path to provide End-to-End QoS over Stateless Networks by Means of a Probing-driven Admission Control

Status of this Memo

1 Abstract

This document proposes an admission control paradigm, called GRIP (Gauge&Gate Reservation with Independent Probing), devised to transparently operate over DiffServ domains. GRIP relies the decision to admit a new flow upon the successful and time delivery, through the Internet, of probe packets independently generated by the end points. The key idea is to use failed receptions of probes to discover, at the end points, that a congestion condition occurs in the network, and to reject the new

admission request. This idea is extremely close to what TCP congestion control technique does, but it is used in the novel context of admission control. While GRIP can be seamlessly applied to DiffServ (and even legacy) Internet, a marginal increase in QoS is envisioned in these existing scenarios. The performances of GRIP are in fact related to the capability of routers to locally take decisions about the degree of congestion in the network, and suitably drop probe packets when congestion conditions are detected.

GRIP can be applied in a "decoupled" framework where admission control is categorized as:

- End-to-end, where the end points of the admission control; procedure are two end hosts;
- Cross-domain, or inter-domain, where the end points of the reservation are located in different administrative domains but not on end hosts;
- Edge-to-edge, or intra-domain where the end points of the admission control procedure are two edge nodes located in the same administrative domain.

Finally, we are fully aware that the possible application of the principles described in this draft in the Internet raises many issues, which we do not address.

Our aim, then, is not proposing a full-fledged solution for the

Internet, but contributing to the on-going discussions in the international arena on these matters, by means of what we may see as a problem statement document.

Table of Contents

2 Introduction

Two QoS architectures are being discussed in the Internet arena: Integrated Services and Differentiated Services. Nevertheless, quoting the recent RFC [RFC2990], "both the Integrated Services architecture and the Differentiated Services architecture have some critical elements in terms of their current definition, which appear to be acting as deterrents to widespread deployment... There appears to be no single comprehensive service environment that possesses

both service accuracy and scaling properties". Our agreement with the above statement is motivated as follows.

The IntServ/RSVP paradigm [RFC2205, RFC2210] is devised to establish reservations at each router along a new connection path, and provide "hard" QoS guarantees. The common criticism to RSVP is related to its complexity and lack of scalability. In the heart of large-scale networks, the cost of RSVP soft state maintenance and of processing and signaling overhead in the routers is significant.

Moreover, we argue that complexity and scalability are not the unique problem of RSVP. RSVP needs to be deployed in all the involved routers, to provide end-to-end QoS guarantees; hence this approach is not easily and smoothly compatible with existing infrastructures. What we are trying to say is that complexity and scalability are really important issues, but that backward compatibility and smooth Internet upgrade in a multi-domain Internet market scenario is probably even more important.

Following this line of reasoning, we argue that the success of the DiffServ framework [RFC2474, RFC2475] does not uniquely stays in the fact that it is an approach devised to overcome the scalability limits of IntServ. As in the legacy Internet, the DiffServ network is oblivious of individual flows. Each router merely implements a suite of scheduling and buffering mechanisms, to provide different aggregate service assurances to different traffic classes whose packets are accordingly marked with a different value of the Differentiated Services Code Point (DSCP) field in the IP packet header. By leaving untouched the basic Internet principles, DiffServ provides supplementary tools to further move the problem of Internet traffic control up to the definition of suitable pricing/service level agreements (SLAs) between peers. However, DiffServ lacks a standardized admission control scheme, and does not intrinsically solve the problem of controlling congestion in the Internet. Upon overload in a given service class, all flows in that class suffer a potentially harsh degradation of service. RFC [RFC2998] recognizes this problem and points out that "further refinement of the QoS architecture is required to integrate DiffServ network services into an end-to-end service delivery model with the associated task of resource reservation". It is thus suggested [RFC2990] to define an "admission control function which can determine whether to admit a service differentiated flow along the nominated network path".

3 Related work

Recent literature (see [BRE00] and references therein contained) has

shown that such an admission control function can possibly be provided over stateless networks by means of the so-called Endpoint Admission Control (EAC). EAC builds upon the idea that admission control can be managed by pure end-to-end operation, involving only the source and destination host. At connection set-up, each sender-receiver pair starts a Probing phase whose goal is to determine whether the considered connection can be admitted to the network. In some EAC proposals [BOR99, ELE00, BRE00], during the Probing phase, the source node sends packets that reproduce the characteristics (or a subset of them) of the traffic that the source wants to emit through the network. Upon reception of the first probing packet, the destination host starts monitoring probing packets statistics (e.g., loss ratio, probes interarrival times) for a given period of time. At the end of the measurement period and on the basis of suitable

criteria, the receiver takes the decision whether to admit or reject the connection and notifies back this decision to the source node. Although the described scheme looks elegant and promising (it is scalable, it does not involve inner routers), a number of issues come out when we look for QoS performance. A scheme purely based on

endpoint measurements suffers of performance drawbacks mostly related to the necessarily limited (few hundreds of ms, for reasonably bounded call setup times) measurement time spent at the destination. Measurements taken over such a short time and on an end-to-end basis cannot capture stationary network states, and thus the decision whether to admit or reject a call is taken over a snapshot of the network status, which can be quite an unrealistic picture of the network congestion level.

The simplest solution to the above issue (other solutions are being explored, but their complete discussion and understanding is way out of the aims of the present paper) is to attempt to convey more reliable network state information to the edge of the network. Several solutions have been proposed in the literature. [CKN00] proposes to drive EAC decisions from measurements performed on a longer time scale among each ingress/egress pair of nodes within a domain. [GKE99, SZH99, KEL00] use packet marking to convey explicit congestion information to the relevant network nodes in charge of taking admission control decisions. [MOR00] performs admission control at layers above IP (i.e., TCP), by imposing each core router to parse and capture TCP SYN and SYN/ACK segments, and forward such packets only if local congestion conditions allow admission of a new TCP flow. [ALM98] proposes a lightweight signaling protocol, with explicit reservation messages, which requires network routers to actively manage packets (via remarking of signaling packets when congestion occurs), and thus it does not fit within a DiffServ framework, where the core routers duty is strictly limited to forwarding packets at the greatest possible speed (see e.g., what

stated in [BRE00]).

To summarize the above discussion, and to proceed further, we can state that an abstract and general EAC can be defined as the combination of three logically distinct components (although, in some specific solutions the following issues are not clearly distinct, this does not mean at all that these three specific issues are not simultaneously present):

1: edge nodes in charge of taking explicit per flow accept/reject decisions;

2: physical principles and measures on which decisions are based (e.g., congestion status of an internal link or an ingress/egress path, and particular measurement technique - if any - adopted to detect such status);

3: the specific mechanisms adopted to convey internal network information to edge nodes (e.g., received probing bandwidth measurement, IP packet marking, exploitation of layers above IP with a well-defined notion of connection or even explicit signaling).

In such a view, and with reference to each of the above points, we

argue that:

1: to allow edge nodes to take learned accept/reject decisions, the congestion status of the network can not be inferred only on an end-to-end basis; inner routers must be actively involved, but without adding functionality other than that of the DiffServ paradigm, in the basic IP forwarding scheme.

2: inner routers can determine whether a new call can be locally admitted (i.e. as far as the local router is concerned) by means of suitable Measurement Based Admission Controls (MBAC). Such MBAC schemes operate according to some specific criteria (which can be as simple as non performing any measure at all, and taking a snapshot of the link state, or as complex as some of the techniques proposed in [BJS00, GRO99]). These schemes do not exploit per-flow state information and related traffic specifications. Instead, they operate on the basis of per-node aggregate traffic measurements carried out at the packet level. The robustness of these schemes stays in the fact that, in suitable conditions (e.g. flow peak rates small with respect to link capacities), they are barely sensitive to uncertainties on traffic profile parameters. As a consequence, it seems that scalable estimations can be independently carried out by the routers as far as local decisions are concerned. As a matter of fact we propose one of such schemes in [BBFP01].

3: An important problem is then how to convey the status of inner routers to the end points so that the latter devices can take learned admission control decisions, without violating the DiffServ paradigm. For obvious reasons, we cannot use explicit

per flow signaling. Similarly, we do not want to modify the basic router operation, by introducing packet marking schemes or forcing routers to parse and interpret higher layer information. What we want to do is to implicitly convey the status of core routers to the end points, by means of scalable, DiffServ compliant procedures.

4 A "Decoupled" Approach to Admission Control

We feel that the way to QoS provisioning in the Internet should be outlined following an evolutionary approach. For evolutionary approach, we mean that each individual domain should be put in the condition of independently and asynchronously upgrade its network components and management schemes to provide support for QoS.

This implies that the point 3) above must be decoupled in the following elements:

1. Intra-domain resource reservation mechanisms. These mechanisms should be limited to provide admission and congestion control functions whose scope is limited to a single administrative domain, and whose design is related to the specific requirements of the

considered domain (e.g. a radio access network, a core backbone, a small campus LAN, etc). The degree of QoS support provided within each domain will depend on the tightness of control that the edge-to-edge mechanism will be capable to support. Schemes ranging from explicit per-flow resource reservation mechanisms (such as RSVP), down to aggregate forms of traffic control (e.g. via measurement based mechanisms, such as the one of GRIP) should be allowed to independently operate in different domains. The ultimate goal is that each domain should be placed in the ideal condition of determining the suitable throughput/QoS support tradeoff within the domain.

2. Inter-domain signaling mechanisms. To allow heterogeneous domain to exchange basic control information, a cross-domain signaling procedure should be deployed. Our view of such a cross domain signaling exchange is twofold:

a: one possibility is to deploy a novel standard to allow domains to exchange control information (e.g. whether a flow can be admitted in the considered domain). The drawback of such a solution is that the format and the contents of these control packets needs to be standardized, and this may limit the timely deployment of this cross-domain mechanism.

b: a much more simple, and in our opinion, appealing possibility, is to define an IMPLICIT cross-domain signaling scheme, based on

drop of signaling packets. More discussion about this solution is given in section 5 and 6.

5 The Concept of Implicit Signaling and its Use in Admission Control

Implicit signaling has been adopted to control network congestion since the introduction of TCP congestion control in 1986. The idea of implicit signaling is to allow the network endpoints to autonomously determine whether congestion occurs along the network path, and to react accordingly.

Congestion conditions are discovered at the end points by analyzing packet losses. Upon congestion within a network node, packets are lost, and this information is implicitly conveyed to the end nodes.

In particular, the authors of this draft have recently proposed an implicit signaling paradigm, called GRIP (Gauge&Gate Reservation with Independent Probing), devised to be compatible with DiffServ scenarios [BB01, BBFP01]. GRIP is DiffServ compliant since all traffic is managed according to the DS Code Point field only. In particular, [BB01] shows that the GRIP way of operation is semantically compatible with the AF PHB [RFC2597]. GRIP is briefly described below.

5.1 GRIP End nodes operation

GRIP's end nodes operation is extremely simple. Let us consider the setup of an "uplink" (source to destination) monodirectional flow. When a user terminal requests a connection with a destination terminal, the Source Node starts a Probing Phase, by injecting in the network in principle just one Probe Packet. Meanwhile, it activates a probing phase timeout, lasting for a reasonably low time. If no response is received from the destination node before the timeout expiration, the source node enforces rejection of the connection setup attempt. Otherwise, if a Feedback packet is received in time, the connection is accepted, the probing phase is terminated, and control is given back to the user application which starts a Data Phase, simply consisting in the transmission of information packets.

The role of the Destination Node simply consists in monitoring the incoming IP packets, intercepting the ones labeled as Probes, reading their source address, and, for each incoming probe packet, just relaying with the transmission of a feedback packet, if the destination is willing to accept the set-up request.

The only mandatory requirement is that Probes and Information

packets are labeled with different values of the DS codepoint field in the IP packet header. This enables DiffServ routers to provide different forwarding methods for Probes and Information packets, e.g. granting service priority to Information packets. In this case, the Feedback packet shall be labeled as an Information packet (i.e., prioritary). Probing packets do not carry information describing the characteristics of the associated data traffic (e.g. peak bandwidth). This information is eventually conveyed by means of the DSCP tag (i.e. a given kind of data traffic is associated with a given DSCP tag).

Note that the described GRIP operation is trivially extended to provide setup for bidirectional connections. In such a case, the destination node will simply relay with a Probe packet instead than with a Feedback packet. A Feedback will be ultimately sent back by the source node upon reception of the destination Probe (to close the three way connection setup handshake - independent probing mechanisms are clearly needed to test both uplink and downlink network paths, which generally differ). Finally, GRIP can be adapted to support "downlink" (destination to source) flows. The source node needs to issue a Trigger Packet to drive (by mean of application-level protocol information, contained in the Trigger Packet payload) the destination node to start a Probing Phase on its own.

To protect GRIP from possible route changes, due to the eventual dynamics of routing protocols, we can think to additional Probing

packets periodically sent after the setup of a flow to "refresh" the end-to-end path. On the other side, DiffServ will be probably deployed in the core network where forwarding mechanisms such as

MPLS, will limit the frequency of route changes below typical session duration. Note also that lost or severely delayed probe packets are interpreted as congestion. A probe packet may be lost if the (wireless) link has high error rates, or delayed if retransmission at lower layers occurs. However, this problem is common to other admission control frameworks and can be overcome by defining more complex probing phase operations, e.g., by including reattempt procedures after a setup failure, multiple timers and probes during the probing phase, etc. This could lead to too much extra traffic generated by probes, which is a phenomenon that could occur also for instance with HTTP session where multiple TCP connections are initiated. To alleviate the problem, Probes could be piggybacked on TCP SYN packets.

Finally, we point out that a priority among probing packets belonging to different traffic classes could be introduced by means of different DSCP tags. This way, higher service class users would

receive favorite treatment. Still another issue is re-negotiation of the flow parameters and requested performance after the flow is accepted.

5.2 GRIP over a GRIP-unaware domain

The rationale of GRIP is to reject a new flow setup when a feedback does not return to the source node before that the probing timeout expires. When GRIP is operated over a GRIP-unaware domain, flow rejection is purely driven by internal network congestion. Upon congestion, the round trip delay (Probe plus Feedback) may become larger than the probing phase timeout, and thus a flow setup is rejected. Stability is guaranteed by the fact that, when network congestion increases, a corresponding decrease in the probability that setup is successful occurs. Therefore, a lower number of new flows set up, and this allows the network to smoothly decongest. Routers may be in principle oblivious of Probes, and may treat them as normal IP packets. When packet differentiation is possible, as in the DiffServ scenario, GRIP operation can be enhanced. This particularly occurs when DiffServ routers are configured to distinguish information packets from Probes on the basis of their DSCP value, and serve information packets with higher service priority (i.e. before) than probing packets. This operation has the advantage that the delay experienced by Probing packets is necessarily worse (and thus is a conservative measure) than that experienced by packets belonging to accepted connections. Thus,

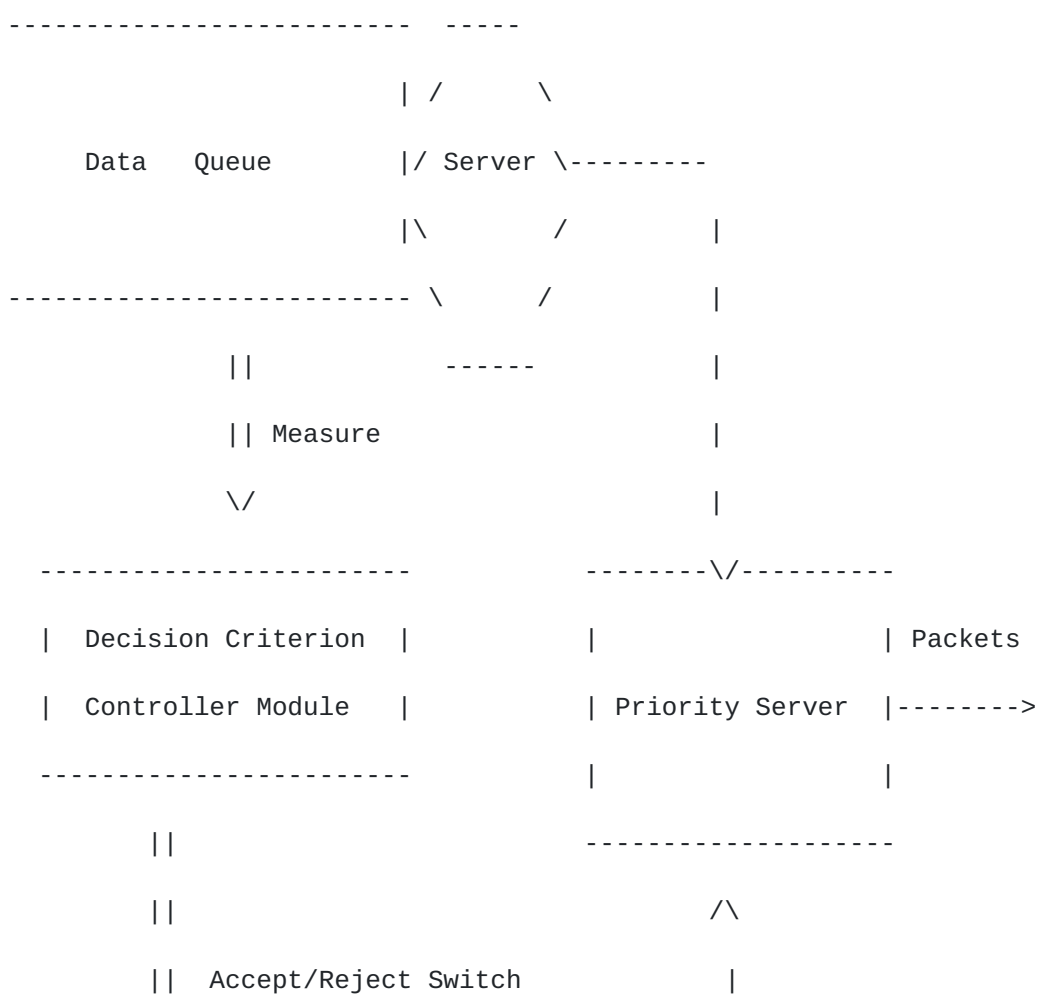
probes may detect internal router congestion earlier than data packets, and earlier drive reject decisions at the end points. The performance of GRIP over DiffServ routers has been preliminarily evaluated in a previous paper of ours. Such results lead to the conclusion that the throughput performance is marginally dependent on the probing packet timeout setting, at least when this timeout is kept in the order of at most few hundreds of ms. This implies that

the probing timeout is not an effective and tunable mean to precisely control the QoS.

5.3 GRIP over a GRIP-aware domain

Despite the above discussed performance drawbacks, our strongest argument in favor of GRIP is that it opens a smooth migration path toward a future QoS capable global infrastructure. Our thesis is that GRIP widespread deployment may start over the actual best-effort Internet to provide marginal performance improvements (i.e. similar to the ones relevant to the Controlled Load service), with the promise that QoS will be provided in the future by independent router upgrades in independent IP domains.

To justify our statement, we assume that network routers are able to recognize that packets labeled as Probes are managed at the network end points for the sake of flow admission control. Hence, they may intelligently enforce Probe dropping, on the basis of suitable estimation of the QoS provided to the already admitted flows, and on the basis of suitable predictions of emerging congestion conditions. As, thanks to the GRIP operation, internal probe losses drive setup rejections at the distributed end points, independent, localized and proprietary decisions taken at the network routers may substantially improve the QoS provided within a domain. The GRIP-aware router operation is illustrated in Fig. 1.



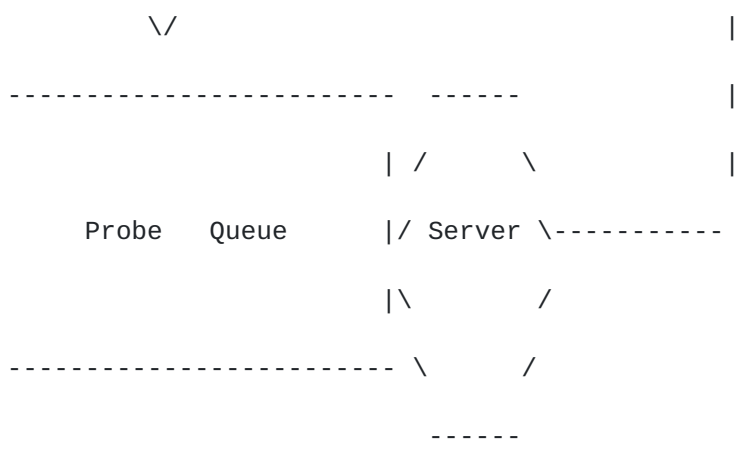


Figure 1: GRIP router operation

For convenience of presentation, we assume that the router handles only GRIP controlled traffic. Other traffic classes (e.g., best-effort traffic) can be handled by means of additional queues, eventually with lower priority. At each router output port, GRIP implements two distinct queues, one for data packets, i.e. belonging to flows that have already passed an admission control test, and one for probing traffic. Packets are dispatched to the respective buffers according to the probe/data DSCP tag. The GRIP router measures the aggregate accepted traffic. On the basis of the running traffic measurements, the router enforces a Decision Criterion,

which continuously drives the router to switch between two states: ACCEPT and REJECT. When in the ACCEPT state, the Probing queue accommodates Probe packets, and serves them according to the described priority mechanism. Conversely, when the router switches to the REJECT state, it discards all the Probing packets contained in the Probing queue, and blocks all new Probing packets arriving. In other words, the router acts as a gate for the probing flow, where the gate is opened or closed on the basis of the traffic estimates (hence the Gauge&Gate in the acronym GRIP).

This mechanism provides an implicit signaling pipe to the end points, of which the network remains unaware. Each router is locally in charge of deciding, on the basis of its own criteria, whether it can admit new flows, or it is congested. The internal router decision is summarized in the router state (ACCEPT vs. REJECT), and it is implicitly advertised to the end points (whose flow setup path crosses the considered router) by letting Probes cross through the router (ACCEPT) or blocking probes (REJECT).

With reference to the performance achievable, it is easy to conclude that the level of QoS support provided depends on the degree of effectiveness of the Decision Criterion implementation. Several Measurement-Based mechanisms [BJS00] have been described in the literature and may be applied to the GRIP routers [e.g., GRO99].

An example of a trivial decision criterion is to accept all Probe

packets when the measured throughput is lower than a given threshold and reject them packets when the measurements overflow this threshold. The resulting delay performance depends upon the link capacity and the traffic model.

Tighter forms of traffic control are possible. As a second example of a decision criterion, we demonstrated that hard (loss and/or delay) QoS guarantees can be provided, within a specific domain, under suitable assumptions on the offered traffic (i.e., traffic sources regulated by standard Dual Leaky Buckets, as in the IntServ framework) and with ad hoc defined measurement modules in the routers [BBFP01].

Finally, we note that the decision criterion must not be necessarily driven by traffic measurements. In fact, it can be driven by lower layers QoS capabilities, (e.g., ATM, MAC) or by tunable proprietary schemes.

A last consideration is that GRIP shares with common MBAC schemes

the problem of defining precise admission control criteria when the admitting flows are very different between each other in their characteristics and in their required performance. To maintain the advantages of GRIP, probes should not contain signaling information to be parsed at core routers (while edge routers could execute this function, see section 6). A possible way to solve this problem is to impose that a given admission controlled traffic class is composed of flows with homogeneous (or at least similar) characteristics and requirements. In other words, QoS enabled sources are divided in traffic classes, each comprising homogeneous (or similar) sources. By envisioning a very small number of traffic classes (e.g., a class could be IP telephony), each class could be handled in a differentiated way, (according to the DiffServ approach, with its own pair of DS codepoints for probing and data), by means of suitable scheduling mechanisms, similar to those already defined (e.g., WFQ, separate queues). Further details on this issue can be found in [BB01].

We conclude by remarking that GRIP does not require any specific protocol implementation in the core routers, which are stateless and remain oblivious to individual flows. Scalability is guaranteed by the fact that (i) no state information is stored in any router, which handle traffic aggregates and not single flows, and that (ii) the whole operation is fully distributed: the procedures have a local scope and each network device operates autonomously.

6 Implicit Cross-Domain Signaling

The principle of packet losses as a way to notify congestion can be extended to heterogeneous domains, each running independent intra-domain reservation mechanisms. The foundation for implicit signaling is only the capability for each ingress node of a domain to recognize whether a packet contains signaling information versus data payload, regardless of which specific signaling information is actually contained. Note that this feature is possible by using suitable packet marking in the DSCP field of the packet header [see also BB01].

To better clarify, consider the scenario depicted in figure 2.

Here, the source to destination path comprises three different domains, namely A, B, and C, each running a different - fictitious - intra-domain reservation protocol (namely RP1, RP2, RPX). Each

reservation protocol has its own scheme, and is triggered by a signaling packet eventually containing suitable information.

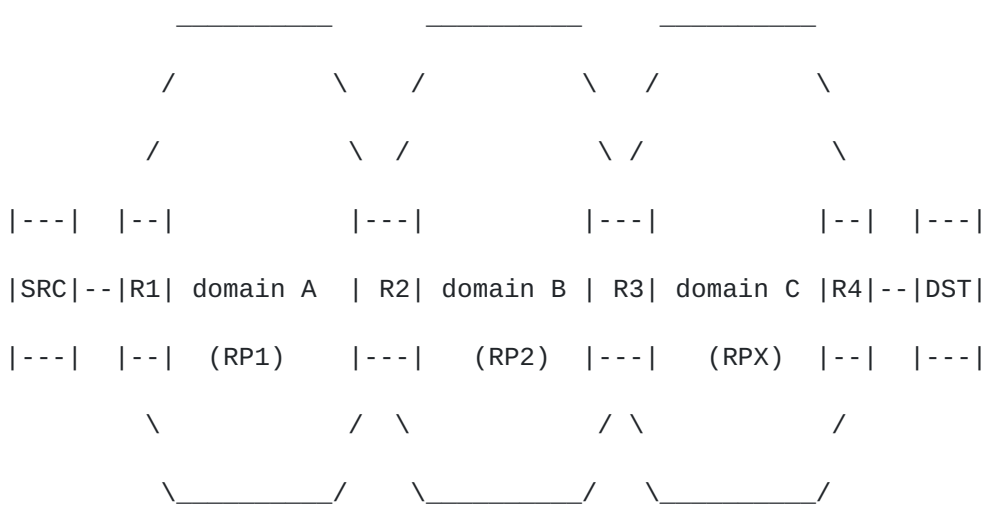


Figure 2: Multi-domain scenario

When the source needs to setup a flow to the destination, it injects in the network a signaling (probe) packet. Similarly to what described above in GRIP, the source node is in charge to wait for a feedback packet, and then activate the flow by emitting data packets. In case the feedback packet does not arrives back in due time, the flow setup is aborted.

The signaling packet injected in the network can carry application-level information to be used at the destination node. In addition, it can eventually carry information that can be read by some reservation protocols, e.g. RP1.

First, the signaling packet arrives at the ingress node of domain A. This node recognizes, in the order, that

a: the packet is a signaling packet, and

b: it contains information usable by the specific reservation protocol RP1 (e.g. RSVP).

This packet thus triggers the specific edge-to-edge reservation mechanism RP1 running through domain A. At the end of the reservation procedure, if the domain is capable of admitting the flow, then the signaling packet is forwarded out of the domain by the egress node. Otherwise, it is dropped.

The same approach is adopted at domain B. Here, the difference is that the specific reservation protocol triggered by the arrival of the signaling packet is different from that adopted in the previous domain (e.g., domain B adopts a DS framework augmented with GRIP admission control functionality, as its inner reservation scheme). However, the result is semantically consistent with the previous domain operation, i.e. the triggering packet is forwarded if the connection can be accepted, and dropped otherwise. No explicit signaling information is exploited, with the exception of the one carried by the DS codepoint of the triggering packet [see BB01].

Finally, the triggering packet arrives at the ingress node of domain C. Here, the ingress node recognizes that the packet is for

signaling, but it finds that the packet does not carry information useful for the reservation protocol RPX (i.e. the packet and even the relevant DS codepoint is incompatible with this domain inner DS procedures). Therefore, domain C can decide to:

a: run a "generic" (e.g. un-parameterized) edge-to-edge signaling procedure

b: drop the packet (i.e. drop the entire flow).

c: simply forward the packet, with no admission control, on a best effort basis.

Although the above example is very loose, and several problems need a thorough investigation, nevertheless it appears that such an implicit signaling approach can be the "glue" for the coexistence of highly heterogeneous edge-to-edge reservation mechanisms.

Moreover, note that the outlined approach allows the coexistence of domains running a reservation protocol with best effort domains. Clearly, the QoS provisioned to the considered end-to-end flow will be bottlenecked by the worst case domain. But in the same time, domains that run a reservation mechanism are capable of limiting the traffic admitted, and thus locally guaranteeing QoS support.

A thorough understanding of this latter issue is of importance. The cross-domain reservation scheme described above is not necessarily aimed at providing an end-to-end QoS support or performance

guarantees. Conversely, it is devised to guarantee each domain that the performance encountered by packets crossing the given domain are kept under control (depending on the degree of tightness of the reservation protocol adopted). In other words, our view of the performance provided is domain-centric, rather than an end-to-end guaranteed performance view. Eventually, suitable routing schemes and SLAs can find a path that comprises only QoS aware domains.

Note that this is line with the way of operation of other functions in the Internet (e.g. routing), which allow different domains to adopt different schemes.

A last issue regards the definition of DS codepoints to identify probe (signaling) packets and data packets. In [BB01] we proposed to use two dropping levels of a given AF class to this purpose. However, we are aware that our suggested usage of AF is different (and quite unexpected) from what intended in RFC 2597. The services that are expected to make use of admission control are RTP/UDP streams with delay and loss performance requirements, whose support is currently envisioned by means of the EF PHB. On the contrary, AF appears designed to provide better than best effort support for generic TCP/UDP traffic. Thus, our study raises the case for the transformation of the (single) EF PHB into a PHB class (i.e. by adding an associated, "paired", probing pipe with a different DSCP). An alternative is defining new "paired" PHBs.

On a different prospective, "paired" PHBs can be envisioned to support more general control functions than admission control. For example, the TCP fast retransmission and recovery algorithm might take advantage of isolated data packets labeled as "control", and thus expected to encounter loss if (controlled) congestion is encountered in the network.

7 Appendix D: Security considerations

As all admission control functions, our solution presents the risk of theft of resources through the unauthorized admission of traffic. Although, logically, user terminals are the natural nodes where the endpoint admission control should operate, this is clearly not realistic, for the obvious reason that the user may bypass the admission control test and directly send probe packets. Identity authentication and integrity protection are therefore needed in order to mitigate this potential for theft of resources [RFC2990]. Administrators are then expected to protect network resources by configuring secure policers at interfaces (e.g. access routers) with untrusted customers. Similar protections must be provided at the

interface between different domains. In particular, it may be necessary to restrict the access to the DS class(es) used for admission controlled traffic. For example, a DS domain should re-mark packets when they come from an un-trusted adjacent DS domain. In more generality, we remark that policing and conditioning rules enforced at the border routers of each domain depend on the usage of the considered class within the specific domain and thus have to be accounted of in the definition of each specific PDB supporting admission control.

A quite obvious security hazard is flooding the network with probe packets. The objective is twofold. On one side, denial of service situations can be easily created, as a massive loading of the network with probe packets prevent the setup of normal connection. On the other side, the goal might be to affect fairness: the continuous transmission of probe packets at a rate higher than normal connection requests is a mean to gain faster access to resources when these are made available by a router along the path. This implies that some form of traffic conditioning and policing is necessary over probe streams. While it is simple to recognize an hard attack, by monitoring the probe packets crossing an edge router (the probe traffic - at most a few packets per originating connection - is minimal in normal conditions, and thus sudden increments of the probe load are suspicious), it may be not straightforward for DS boundary routers to recognize smoother fairness attacks. However, note that the same fairness problem is present also in more complex reservation mechanisms, such as RSVP

(malicious users can continuously require setup to increase their access possibility with respect to normal users).

Finally, all the security considerations expressed in [RFC2990] apply also to our solution.

8 References

[ALM98] W.Almesberger, T.Ferrari, J. Y. Le Boudec: "SRP: a Scalable Resource Reservation Protocol for the Internet", IWQoS'98, Napa (California), May 1998.

[BB01] G. Bianchi, N. Blefari-Melazzi: "Per Flow Admission Control over AF PHB Classes", Internet draft, draft_bianchi_blefari_admcontr_over_af_phb.txt, March 2001, work in progress.

[BBFP01] G. Bianchi, N. Blefari-Melazzi, M. Femminella, F. Pugini: "GRIP: Technical report", work in progress,

(http://conan.diei.unipg.it/netweb/GRIP_tech_rep.pdf).

[BJS00] L. Breslau, S. Jamin, S. Schenker: "Comments on the performance of measurement-based admission control algorithms", IEEE Infocom 2000, Tel-Aviv, March 2000.

[BOR99] F. Borgonovo, A. Capone, L. Fratta, M. Marchese, C. Petrioli, "PCP: A Bandwidth Guaranteed Transport Service for IP networks", IEEE ICC'99, June 1999.

[BRE00] L. Breslau, E. W. Knightly, S. Schenker, I. Stoica, H. Zhang: "Endpoint Admission Control: Architectural Issues and Performance", ACM SIGCOMM 2000, Stockholm, Sweden, August 2000.

[CKN00] C. Cetinkaya, E. Knightly, "Egress Admission Control", Proc. of IEEE Infocom 2000, Tel-Aviv, March 2000.

[ELE00] V. Elek, G. Karlsson, "Admission Control Based on End-to-End Measurements", Proc. of IEEE Infocom 2000, Tel Aviv, Israel, March 2000.

[GKE99] R. J. Gibbens, F. P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network", 16 ITC, Edimburgh, June 1999.

[GRO99] M. Grossglauser, D. N. C. Tse: "A Time-Scale Decomposition Approach to Measurement-Based Admission Control", Proc. of IEEE

Infocom 1999, New York, USA, March 1999.

[KEL00] F. P. Kelly, P. B. Key, S. Zachary: " Distributed Admission Control", IEEE JSAC, Vol. 18, No. 12, December 2000.

[MOR00] R. Mortier, I. Pratt, C. Clark, S. Crosby: "Implicit Admission Control", IEEE JSAC, Vol. 18, No. 12, December 2000.

[RFC2205] R. Braden, L Zhang, S. Berson, S. Herzog, S. Jamin, "ResourceReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC2205, September 1997.

[RFC2210] J. Wroclawsky, "The use of RSVP with IETF Integrated Services", RFC2210, September 1997.

[RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definitions of the Differentiated Service Field (DS Field) in the Ipv4 and Ipv6 Headers", RFC2474, December 1998.

[RFC2475] S. Blade, D. Black, M. Carlson, E. Davies, Z. Wang, W.

Weiss, "An Architecture for Differentiated Services", RFC2475, December 1998.

[RFC2597] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

[RFC2990] G. Huston, "Next Steps for the IP QoS Architecture", RFC2990, November 2000.

[RFC2998] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J. and E. Felstaine, "A Framework for Integrated Services Operation Over DiffServ Networks", RFC 2998, November 2000.

[SZH99] I. Stoica, H. Zhang, "Providing guaranteed services without per flow management", Proc. of ACM SIGCOMM 1999, Cambridge, MA, September 2000.

9 Author's Addresses

Giuseppe Bianchi
DIE, University of Palermo
Viale delle Scienze, Parco d'Orleans
90128 Palermo, ITALY
Tel: +39 091 6566 276
E-mail: bianchi@elet.polimi.it

Nicola Blefari-Melazzi

DIEI, University of Perugia

Via G. Duranti 93, 06125 Perugia, ITALY

Tel: +39 075 585 3630

e-mail: blefari@diei.unipg.it

10 Full Copyright Statement